Recursive Neural Network With Phase-Normalization for Modeling and Linearization of RF Power Amplifiers

Arne Fischer-Bühner, *Graduate Student Member, IEEE*, Lauri Anttila, *Member, IEEE*,
Manil Dev Gomony, *Member, IEEE*, and Mikko Valkama, *Fellow, IEEE*

Abstract— This letter presents a novel phase-normalized recurrent neural network (PN-RNN) to linearize radio frequency (RF) power amplifiers (PAs) in high-bandwidth communication systems with significant memory effects. The proposed approach builds on proper phase alignment of the internal hidden variables in the recursive processing system. The provided RF measurement-based modeling and digital predistortion (DPD) results at 1.8 and 3.5 GHz demonstrate a significantly improved modeling capacity and predistortion ability when applying phase normalization, confirming the validity of the proposed approach.

Index Terms— Behavioral modeling, digital predistortion (DPD), power amplifier (PA), recurrent neural network (RNN).

I. INTRODUCTION

DIGITAL predistortion (DPD) is an established technique to correct for nonlinear impairments of radio frequency (RF) power amplifiers (PAs) in base-station transmitters. While PAs are bound to efficiency versus linearity tradeoffs, DPD allows operating the PA at a higher efficiency [1] by recovering the required linearity through signal processing. In the upcoming 6G era, signal quality will be of crucial importance for enabling high-throughput communications, with very large instantaneous transmission bandwidths [2]. This challenges existing DPD methods, as PA memory effects occurring with such high signal bandwidths are difficult to compensate for.

Neural networks (NNs) have been identified as a capable and scalable approach for modeling and predistortion of RF PA systems with sophisticated memory effects. Feed-forward NNs such as the real-valued time-delay NN (RVTDNN) have been considered in [3]. While the feed-forward NN itself has no notion of time, a finite history of past samples is provided as input, which enables the modeling of memory effects. For real-valued NN processing, the baseband samples

are decomposed into I and Q parts. In [4], phase normalization is applied to the time-delay inputs (PN-TDNN), which significantly enhances the modeling capability of the RVTDNN. However, higher sampling rates and bandwidths increase the relevant sample history for properly handling the PA memory effects. A finite sample history may constrain the PN-TDNN's modeling ability, whereas inputting more past samples naturally increases the PN-TDNN's complexity.

Despite their popularity in time-series processing, recurrent NNs (RNNs) have been less explored in the DPD context. Unlike feed-forward NNs, RNNs can encode and remember past information with hidden states. Thus, the RNN can identify the relevant history during training, which optimizes the tradeoff between NN size and accuracy. Long short-term-memory (LSTM) RNNs were proposed for DPD in [5] and [6], where only the complex envelope is shaped by the RNN. In [7], a combined approach of LSTM with polynomial kernels is presented. An RNN with improved envelope and phase modeling ability was proposed in [8] and refined in [9].

Building upon the prior literature, we propose the phase-normalized RNN (PN-RNN) by incorporating efficient phase normalization into the recursive processing. We apply phase renormalization to the RNN's hidden states, which we interpret as complex-valued signals with a phase coherent to the NN input. We demonstrate the effectiveness of the proposed PN-RNN with PA modeling and PA linearization experiments and provide a comparison with existing RNN approaches in the DPD literature. Section II introduces the proposed concepts, Section III presents the modeling and linearization results, and Section IV concludes this letter.

II. PROPOSED RNN MODEL

For modeling the PA distortions at the baseband, the NN needs to map the complex-valued input samples $x(k)$ to the corresponding output samples $y(k)$. A top-level block diagram of the proposed PN-RNN is shown in Fig. 1. At the core of the NN is a recurrent layer, composed of N RNN cells, which encode the inputs $X(k)$ into N complex-valued hidden states $H(k)$. The hidden states are looped back so that the states of the previous time step $H(k-1)$ serve as additional inputs to the recurrent layer. The model's output is then formed by a linear combination of hidden states in the output layer. Although any information on past samples can generally be stored in the hidden states of the recurrent layer, we also input a few previous samples alongside the current sample, which we summarize in $X(k) = [x(k), \ldots, x(k-M)]$. These fixed memory inputs are optional—the RNN will also function without them, however, as our experiments will show a slight performance benefit, and a more stable convergence is achieved with the additional inputs.
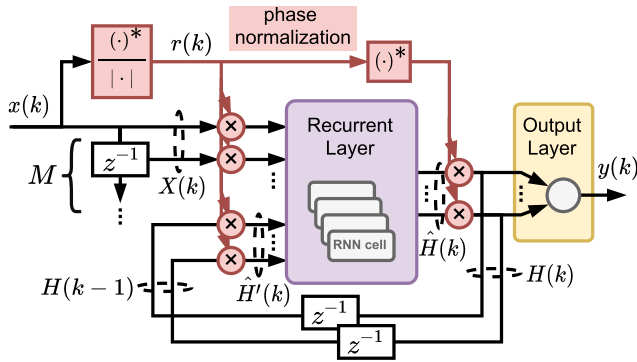
Fig. 1. Illustration of the proposed RNN with phase normalization.

A. Phase Normalization

Similar to the RVTDNN, the NN structures internal to the RNN are limited to real-valued operation. Thus, for the actual NN processing, the complex-valued input and output signals are typically separated into respective I and Q parts. However, as detailed in [4] and [10], the real-valued NN structures cannot efficiently handle the role of the baseband signal phase. That is, PA distortions occurring at RF are independent of the specific baseband phase but depend on the baseband envelope and the phase derivative that modulates the RF carrier frequency. The lack of support for complex-valued operation in real-valued NNs prevents the NN from efficiently expressing passband distortions at the baseband. Instead, their high generality allows us to model effects that are nonphysical at RF. Therefore, [4] introduce phase normalization to remove the specific phase offset from the NN inputs and outputs, and let the NN map distortions based on phase differences relative to a normalization reference. This effectively reduces the NN's generality and helps to comply with the physical origin of the distortions, yielding a significantly improved modeling ability.

Extending the concept of phase normalization to RNNs, also the hidden states need to be considered in the normalization. Since the hidden states carry information across multiple NN inferences, phase normalization cannot be applied to just the NN inputs and outputs. To this end, we propose to include the hidden states in the normalization. As a first step, we define the hidden states to be complex-valued signals that are coherent with the phase of the currently processed sample. Naturally, when applying phase normalization on the NN input $X(k)$, the hidden states will have a phase relative to the normalization reference. As the specific phase changes from one processed sample to another, a different normalization $r(k)$ is applied to each set of inputs and outputs. To then retain a valid phase of the hidden states, we re-normalize the hidden states by complex rotation with the phase difference between two consecutive samples. Formalizing the concept, the normalization factor of the current sample is extracted with

$$r(k) = x^*(k)/|x(k)| \tag{1}$$

with x^* being the complex conjugate of x. Then, the input vector $X(k)$ is phase normalized to $\hat{X}(k)$ through

$$\hat{X}(k) = r(k)\,X(k). \tag{2}$$

The recurrent layer then produces a vector of phase-normalized hidden states $\hat{H}(k)$ and the model output is formed with

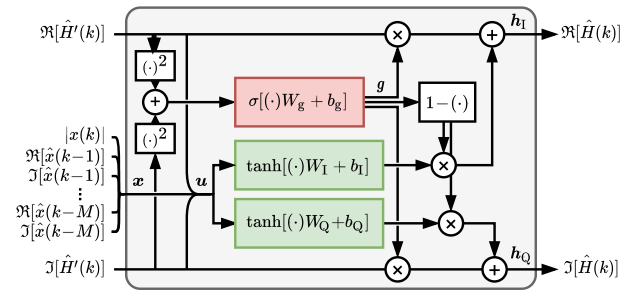$$y(k) = H(k)W_o = r^*(k)\hat{H}(k)W_o \tag{3}$$

Fig. 2. Block diagram of the JANET-inspired RNN cell using dedicated input activations (green) for the I and Q parts and a joint forget gate (red).

where multiplication with $r^*(k)$ denormalizes the phase and $W_o \in \mathbb{C}^N$ are the complex-valued weights of the output layer. The hidden states of the previous iteration are renormalized and fed back into the recurrent layer as inputs $\hat{H}'(k)$ with

$$\hat{H}'(k) = r(k)r^*(k-1)\,\hat{H}(k-1) \tag{4}$$

where $r(k)r^*(k-1)$ equals the phase difference of the two consecutive samples.

B. RNN Cell

For the RNN cells, we take inspiration from [8] and [9]. In line with the PA physics, *Just Another Network* (JANET) [11] was identified as a suitable, lightweight RNN cell for DPD in [8]. Different from the LSTM cell used in [5], a JANET cell consists of only a single sigmoid-based (σ) gating mechanism (forget gate), combined with the hyperbolic input activation (tanh). The JANET concept is refined in [9] with a more tailored RNN cell (DVR-JANET), with additional dedicated filters for phase and envelope, based on the DVR concept [12]. In addition, separate hidden states are introduced for the I and Q parts, which are jointly gated with a common forget gate.

The phase normalization of our proposed RNN eliminates the need for dedicated envelope and phase processing. Instead, we return to a simpler realization of JANET shown in Fig. 2. We maintain the idea of separate hidden states, $\boldsymbol{h}_I$ and $\boldsymbol{h}_Q$ for the I and Q components, which, outside the RNN cell, are considered as complex-valued state with $\hat{H}(k) = \boldsymbol{h}_I + j\boldsymbol{h}_Q$. For the JANET forget gate, we use the squared magnitude of the hidden states, to avoid including the square root inside the trained NN. It is noted that all internal operations and signals of the recurrent cell are real-valued, which is highlighted by bold symbol notation. Additionally, we omit the sample dependency k for notational simplicity. The RNN cell is then given by

$$\begin{cases} \boldsymbol{g} = \sigma\!\left[\left(\boldsymbol{h}_I'^{\,2} + \boldsymbol{h}_Q'^{\,2}\right)W_g + \boldsymbol{b}_g\right] \\ \boldsymbol{u} = \left[\boldsymbol{x},\ \boldsymbol{h}_I',\ \boldsymbol{h}_Q'\right] \\ \boldsymbol{h}_I = (1-\boldsymbol{g}) \otimes \tanh[\boldsymbol{u}W_I + \boldsymbol{b}_I] + \boldsymbol{g} \otimes \boldsymbol{h}_I' \\ \boldsymbol{h}_Q = (1-\boldsymbol{g}) \otimes \tanh[\boldsymbol{u}W_Q + \boldsymbol{b}_Q] + \boldsymbol{g} \otimes \boldsymbol{h}_Q' \end{cases} \tag{5}$$

where $\boldsymbol{h}_I' = \Re[\hat{H}'(k)]$ and $\boldsymbol{h}_Q' = \Im[\hat{H}'(k)]$, and $\otimes$ denotes an elementwise product. The vector $\boldsymbol{x}$ contains the separated real and imaginary parts of the phase normalized inputs $\hat{X}(k)$.

III. MODELING AND RF MEASUREMENT RESULTS

This section provides a performance evaluation of the proposed PN-RNN. For a comparative analysis, we also included
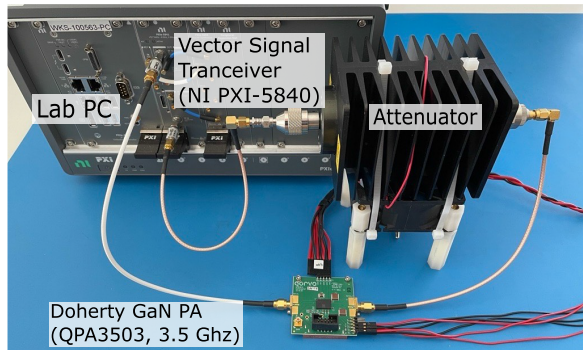
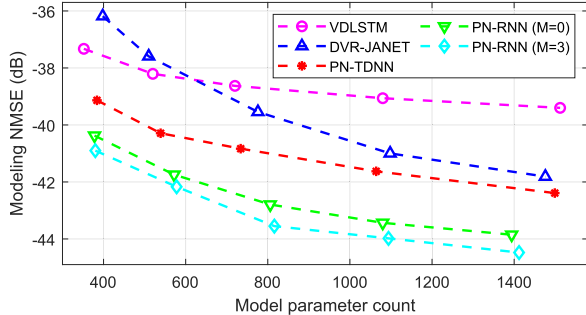Fig. 3.   Measurement setup with VST *(NI PXIe-5840)*, and DUT *(QPA3503)*, attenuator, and laboratory PC.

Fig. 4.   NMSEs of the compared models for modeling the 1.8-GHz GaN Doherty PA.
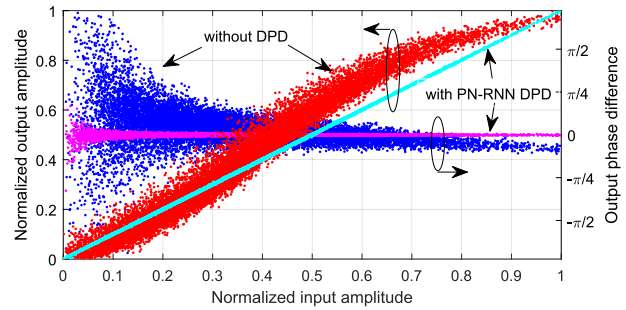

Fig. 5.   AM/AM and AM/PM with and without PN-RNN linearization.
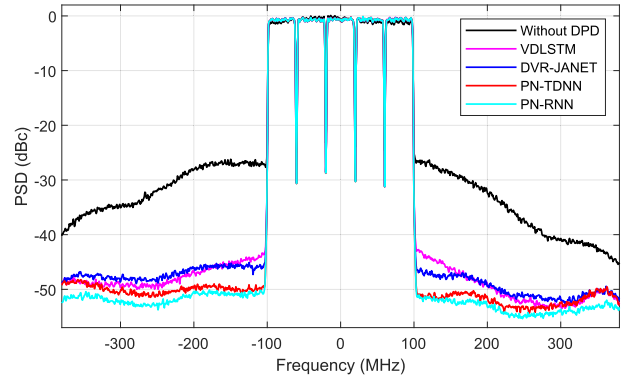

Fig. 6.   PSD of the linearized output of a 3.5-GHz Doherty GaN PA operating a 200-MHz wide multicarrier OFDM waveform. The different RNN models shown include approximately 1k parameters.

the VDLSTM (with four direct memory inputs), the DVR-JANET (three DVR kernels), and the PN-TDNN (two hidden layers) in our evaluation. For all models, we investigate the PA behavioral modeling capacity and the DPD linearization ability. The modeling accuracy is assessed in terms of normalized mean squared error (NMSE). For the DPD linearization, we additionally report the worst adjacent channel leakage ratio (ACLR) and error vector magnitude (EVM). The RF measurements were conducted with the *NI PXIe-5840* vector signal transceiver (VST), which we used for analog signal generation and upconversion to RF, in the transmit path, as well as for the downconversion and digitization of the PA output signal in the DPD feedback path. The PA DUT is mounted on an evaluation board which is connected to the VST in line with a power attenuator. The setup used in the linearization experiment is shown in Fig. 3.

The models were implemented and trained using the *TensorFlow* framework [13]. We optimize the RNN models with a sequence-to-sequence learning approach. Therein, the RNN structures are unfolded during the training phase such that $T = 40$ time steps are processed at the same time. This allows us to estimate coefficient gradients with backpropagation across multiple RNN iterations. The gradients were accumulated for batches of $20T$ samples, after which the model parameters were updated using the Adam optimization algorithm. The models were trained during 2200 repetitions (epochs) on the training dataset, with a decaying learning rate such that a proper convergence of the parameters was achieved.

A. Behavioral Modeling Example

We first assess the forward modeling capability of the models with measured data from a Doherty GaN PA (*RTH18008S-30*) running a 5G-compliant 100-MHz OFDM waveform at 1.8 GHz with an output power of +38.1 dBm.

We used 180k input-output sample pairs for training and dedicated 120k samples for evaluation. The modeling results are shown in Fig. 4, where the complexity of the RNN models is varied by altering the number of hidden states. The data points represent the averaged performance of five training cycles, to account for variations due to the random NN coefficient initialization. For the PN-TDNN we adjusted the sizes of the hidden layers as well as the delay depth of the time-delay inputs. We provide PN-RNN results for $M = 0$ and $M = 3$ direct memory inputs. Both PN-RNN realizations achieve a substantially reduced modeling error, that is, improved accuracy, compared to the reference methods. Additionally, a slight advantage is found when including the direct memory inputs.

B. DPD Linearization Example

Next, we evaluated the linearization capability of the models in a DPD linearization experiment with another GaN Doherty PA (*QPA3503*). A 5G-compliant, continuously aggregated, 200-MHz wide multicarrier waveform with an overall limited PAPR of 8 dB at a center frequency of 3.5 GHz and with a linearized PA output power of +34.2 dBm was applied in the experiment. To generate the training data, we applied the iterative learning control (ILC) scheme from [14] to derive an ideal DPD signal. The RNN parameters were then optimized offline while refreshing the ILC training data after every 400 training epochs. For the DPD evaluation, the output layers of the trained NN models were again fine-tuned and the achieved linearity was assessed with a dedicated test sequence with 210k samples. Fig. 5 shows the AM/AM and AM/PM behavior with and without DPD, and Fig. 6 provides the power spectral density (PSD) of the linearized PA output using the different

TABLE I
DPD EXPERIMENTAL RESULTS WITH THE DOHERTY GaN PA AT 3.5 GHz

Model	Parameters	States	NMSE	ACLR	EVM
Without DPD	–	–	–20.2 dB	–26.1 dB	8.4 %
ILC baseline	–	–	–51.9 dB	–57.9 dB	1.81 %
VDLSTM [5]	1080	15	–36.7 dB	–43.1 dB	2.18 %
DVR-JANET [9]	1098	12	–38.3 dB	–45.0 dB	2.03 %
PN-TDNN [4]	1035	–	–40.8 dB	–49.1 dB	1.94 %
PN-RNN	**1094**	**13**	**–42.8** dB	**–50.0 dB**	**1.89 %**

methods. The corresponding model configuration and detailed results are reported in Table I. The reported PN-TDNN in this experiment was configured with eight time-delay inputs and two hidden layers with 24 and 15 neurons, respectively.

The linearization experiment confirms the trend observed in the behavioral modeling example. The PN-RNN achieves the highest linearity rating in terms of PA output NMSE and ACLR. Also, the EVM approaches the PAPR-limiter-induced minimum.

IV. CONCLUSION

This letter presented a PN-RNN model for modeling and linearization of RF PAs with strong memory effects. Extending the concept of phase normalization to RNNs, not only the RNN inputs and output but also the complex-valued hidden states are considered for phase normalization. This enables the RNN to incorporate phase normalization and retain a valid memory of the I/Q phase trajectory. Additionally, direct memory inputs were studied and shown to further improve the RNN's modeling capability. Our results show a superior behavioral modeling accuracy of the PN-RNN by about 3 dB lower NMSE compared to existing RNN models, which translated to an improved DPD ability with lower ACLR in our PA linearization experiments.

REFERENCES

[1] J. Wood, "System-level design considerations for digital pre-distortion of wireless base station transmitters," *IEEE Trans. Microw. Theory Techn.*, vol. 65, no. 5, pp. 1880–1890, May 2017.

[2] K. Chuang, H. Yektaii, N. Outaleb, A. Raslan, S. Bhal, and P. Forbes, "Towards sustainable networks: Attacking energy consumption in wireless infrastructure with novel technologies," *IEEE Microw. Mag.*, vol. 24, no. 12, pp. 44–59, Dec. 2023.

[3] M. Rawat, K. Rawat, and F. M. Ghannouchi, "Adaptive digital predistortion of wireless power amplifiers/transmitters using dynamic real-valued focused time-delay line neural networks," *IEEE Trans. Microw. Theory Techn.*, vol. 58, no. 1, pp. 95–104, Jan. 2010.

[4] A. Fischer-Bühner, L. Anttila, M. D. Gomony, and M. Valkama, "Phase-normalized neural network for linearization of RF power amplifiers," *IEEE Microw. Wireless Technol. Lett.*, vol. 33, no. 9, pp. 1357–1360, Apr. 2023.

[5] H. Li, Y. Zhang, G. Li, and F. Liu, "Vector decomposed long short-term memory model for behavioral modeling and digital predistortion for wideband RF power amplifiers," *IEEE Access*, vol. 8, pp. 63780–63789, 2020.

[6] G. Xu, H. Yu, C. Hua, and T. Liu, "Chebyshev polynomial-LSTM model for 5G millimeter-wave power amplifier linearization," *IEEE Microw. Wireless Compon. Lett.*, vol. 32, no. 6, pp. 611–614, Jun. 2022.

[7] Q. Zhang, C. Jiang, G. Yang, R. Han, and F. Liu, "Multi-output recurrent neural network behavioral model for digital predistortion of RF power amplifiers," *IEEE Microw. Wireless Technol. Lett.*, vol. 33, no. 7, pp. 1067–1070, Jul. 2023.

[8] T. Kobal, Y. Li, X. Wang, and A. Zhu, "Digital predistortion of RF power amplifiers with phase-gated recurrent neural networks," *IEEE Trans. Microw. Theory Techn.*, vol. 70, no. 6, pp. 3291–3299, Jun. 2022.

[9] T. Kobal and A. Zhu, "Digital predistortion of RF power amplifiers with decomposed vector rotation-based recurrent neural networks," *IEEE Trans. Microw. Theory Techn.*, vol. 70, no. 11, pp. 4900–4909, Nov. 2022.

[10] E. G. Lima, T. R. Cunha, and J. C. Pedro, "A physically meaningful neural network behavioral model for wireless transmitters exhibiting PM-AM/PM-PM distortions," *IEEE Trans. Microw. Theory Techn.*, vol. 59, no. 12, pp. 3512–3521, Dec. 2011.

[11] J. Van Der Westhuizen and J. Lasenby, "The unreasonable effectiveness of the forget gate," in *Proc. Neural Evol. Comp.*, Sep. 2018, pp. 1–15.

[12] A. Zhu, "Decomposed vector rotation-based behavioral modeling for digital predistortion of RF power amplifiers," *IEEE Trans. Microw. Theory Techn.*, vol. 63, no. 2, pp. 737–744, Feb. 2015.

[13] M. Abadi et al. (2016). *TensorFlow: Large-scale Machine Learning on Heterogeneous Systems*. [Online]. Available: https://www.tensorflow.org/

[14] J. Chani-Cahuana, P. N. Landin, C. Fager, and T. Eriksson, "Iterative learning control for RF power amplifier linearization," *IEEE Trans. Microw. Theory Techn.*, vol. 64, no. 9, pp. 2778–2789, Sep. 2016.